US 20040148380A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2004/0148380 A1

Meyer et al. (43) Pub. Date: Jul. 29, 2004

(54) METHOD AND SYSTEM FOR DYNAMIC EXPANSION AND CONTRACTION OF NODES IN A STORAGE AREA NETWORK

(76) Inventors: **Richard Meyer**, San Francisco, CA (US); **Yuri Khodosh**, Sunnyvale, CA (US); **Andrey Gusev**, Fremont, CA (US)

Correspondence Address:
Law Office Of Leland Wiesner
1144 Fife Ave.
Palo Alto, CA 94301 (US)

(21) Appl. No.: **10/695,887**

(22) Filed: **Oct. 28, 2003**

Related U.S. Application Data

(60) Provisional application No. 60/422,109, filed on Oct. 28, 2002.

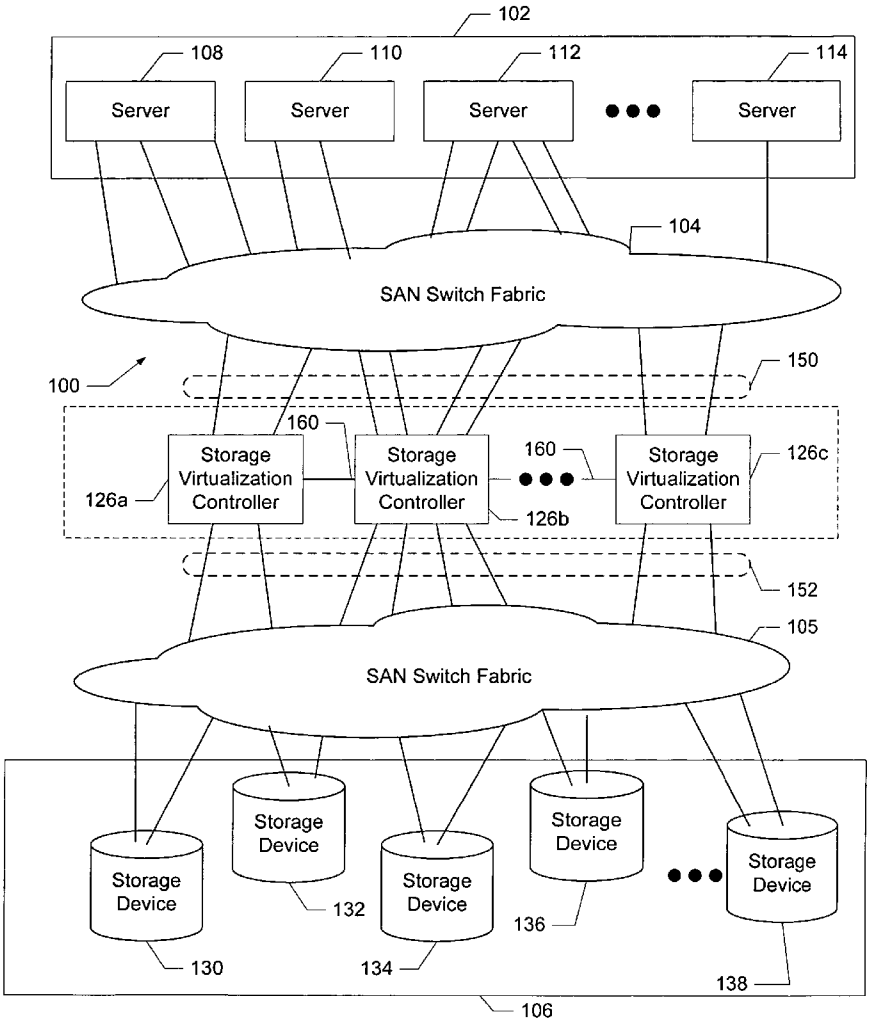
Publication Classification

(51) **Int. Cl.**[7] **G06F 15/173**

(52) **U.S. Cl.** **709/223**

(57) **ABSTRACT**

A method and system to add a storage controller in a SAN includes receiving a storage controller node to add to a logical storage controller in the storage area network having a logical nodename and a sequence of logical ports, adopting the logical nodename from the logical storage controller in place of the predetermined nodename associated with the storage controller and renumbering a set of ports associated with the storage controller to extend the sequence of logical ports associated with the logical storage controller. Removing the storage controller includes removing a storage controller node from the logical storage controller in the storage area network having a logical nodename and a sequence of logical ports, deleting the set of ports associated with the removed storage controller from the sequence of logical ports associated with the logical storage controller and renumbering the sequence of logical ports to accommodate the deleted ports.



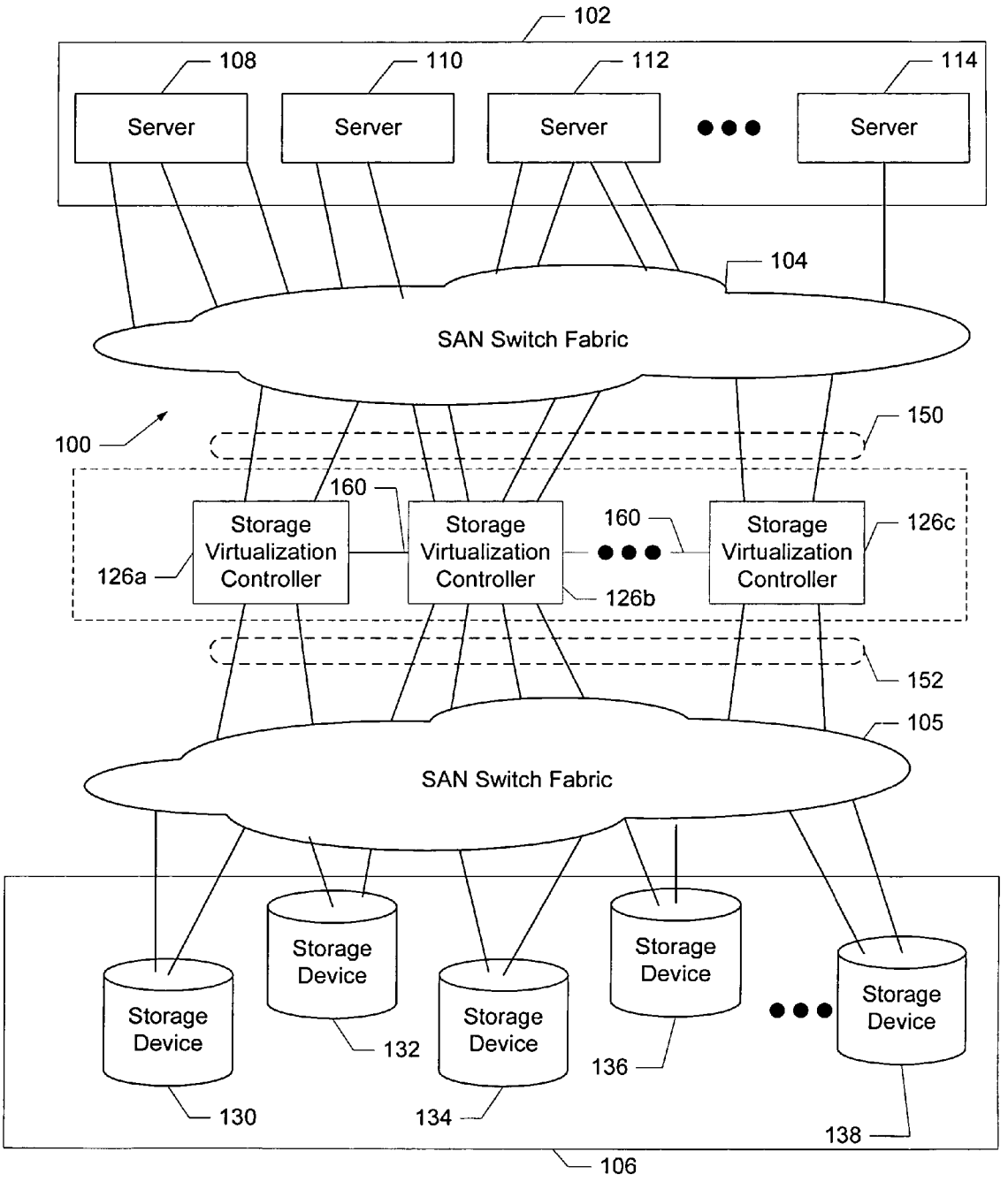

102
108
110
112
114
Server
Server
Server
Server
104
SAN Switch Fabric
100
150
160
160
Storage Virtualization Controller
Storage Virtualization Controller
Storage Virtualization Controller
126a
126b
126c
152
105
SAN Switch Fabric
Storage Device
Storage Device
Storage Device
Storage Device
Storage Device
130
132
134
136
138
106

FIG. 1

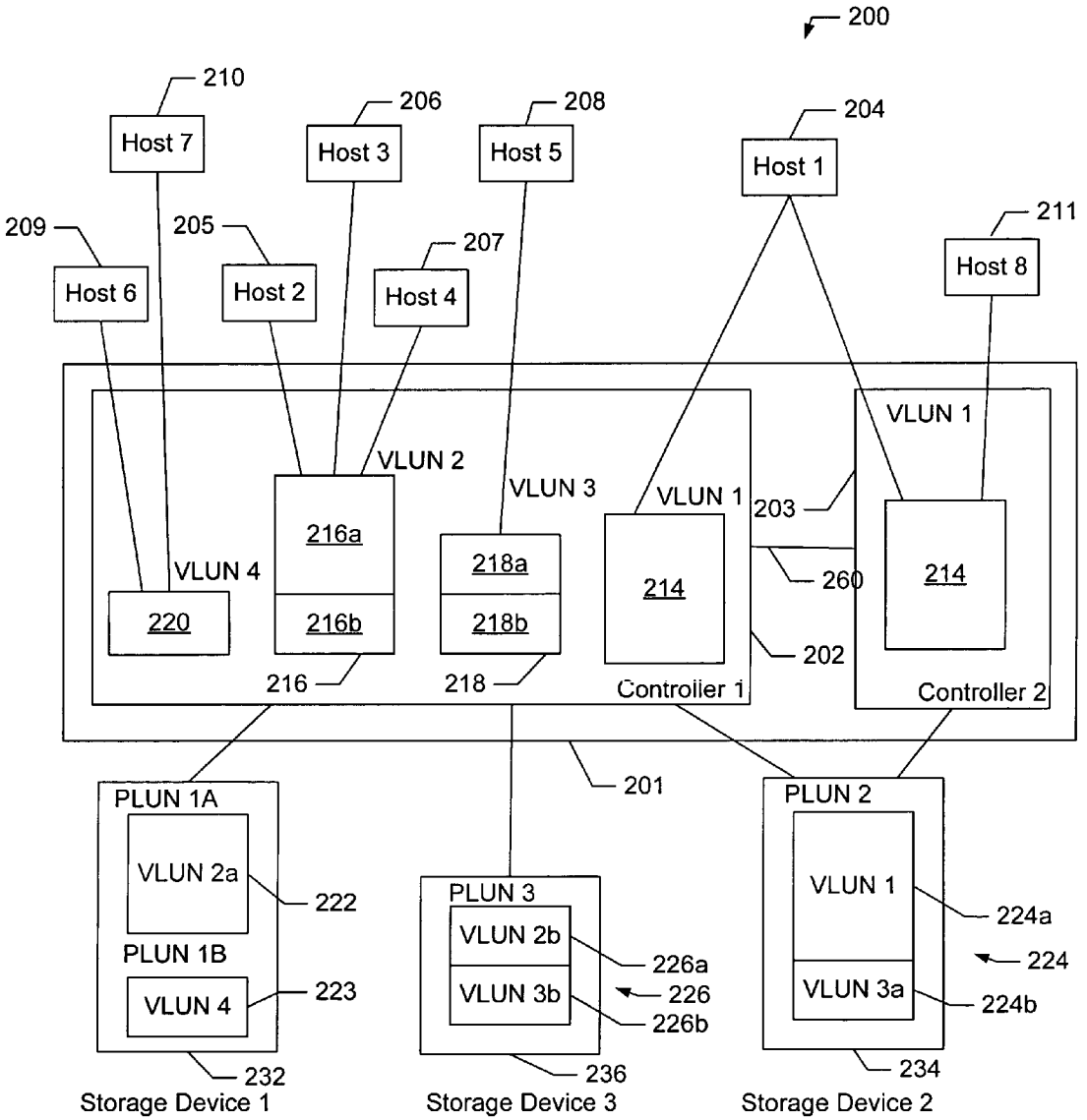

200
210
Host 7
206
Host 3
208
Host 5
204
Host 1
211
Host 8
209
Host 6
205
Host 2
207
Host 4
VLUN 2
VLUN 3
VLUN 1
VLUN 1
203
216a
218a
214
214
VLUN 4
260
220
216b
218b
202
216
218
Controller 1
Controller 2
201
PLUN 1A
VLUN 2a
222
PLUN 1B
VLUN 4
223
232
Storage Device 1
PLUN 3
VLUN 2b
226a
VLUN 3b
226
226b
236
Storage Device 3
PLUN 2
VLUN 1
224a
224
VLUN 3a
224b
234
Storage Device 2

FIG. 2

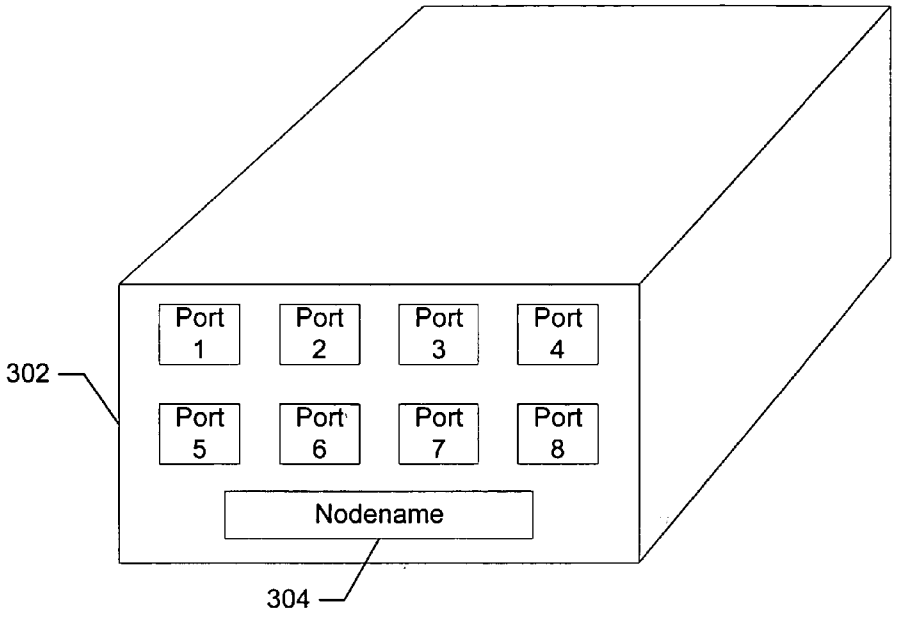

Port 1
Port 2
Port 3
Port 4
Port 5
Port 6
Port 7
Port 8
Nodename
302
304

FIG. 3A

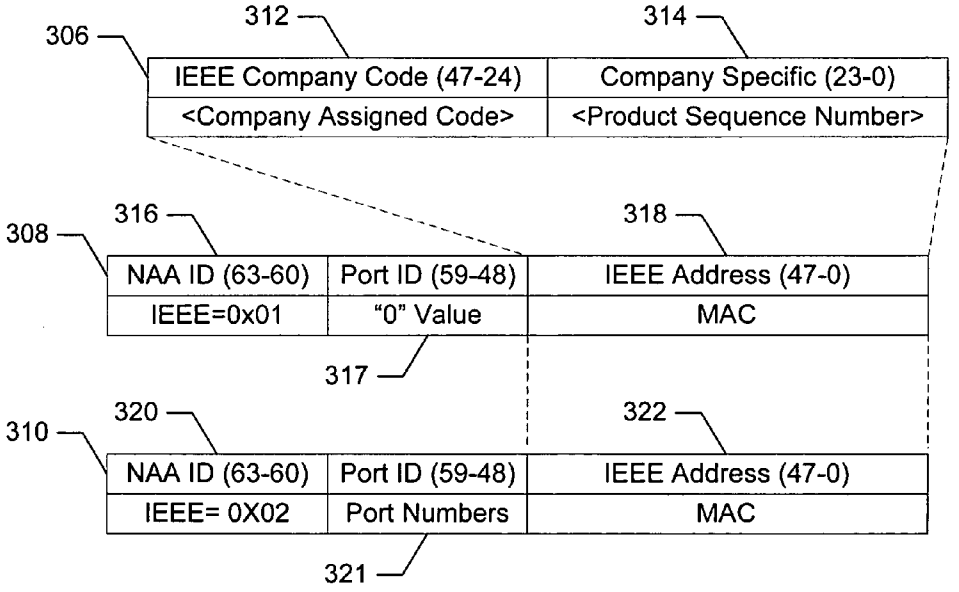

306
312
314
IEEE Company Code (47-24)
Company Specific (23-0)
<Company Assigned Code>
<Product Sequence Number>
308
316
318
NAA ID (63-60)
Port ID (59-48)
IEEE Address (47-0)
IEEE=0x01
"0" Value
MAC
317
310
320
322
NAA ID (63-60)
Port ID (59-48)
IEEE Address (47-0)
IEEE= 0X02
Port Numbers
MAC
321

FIG. 3B

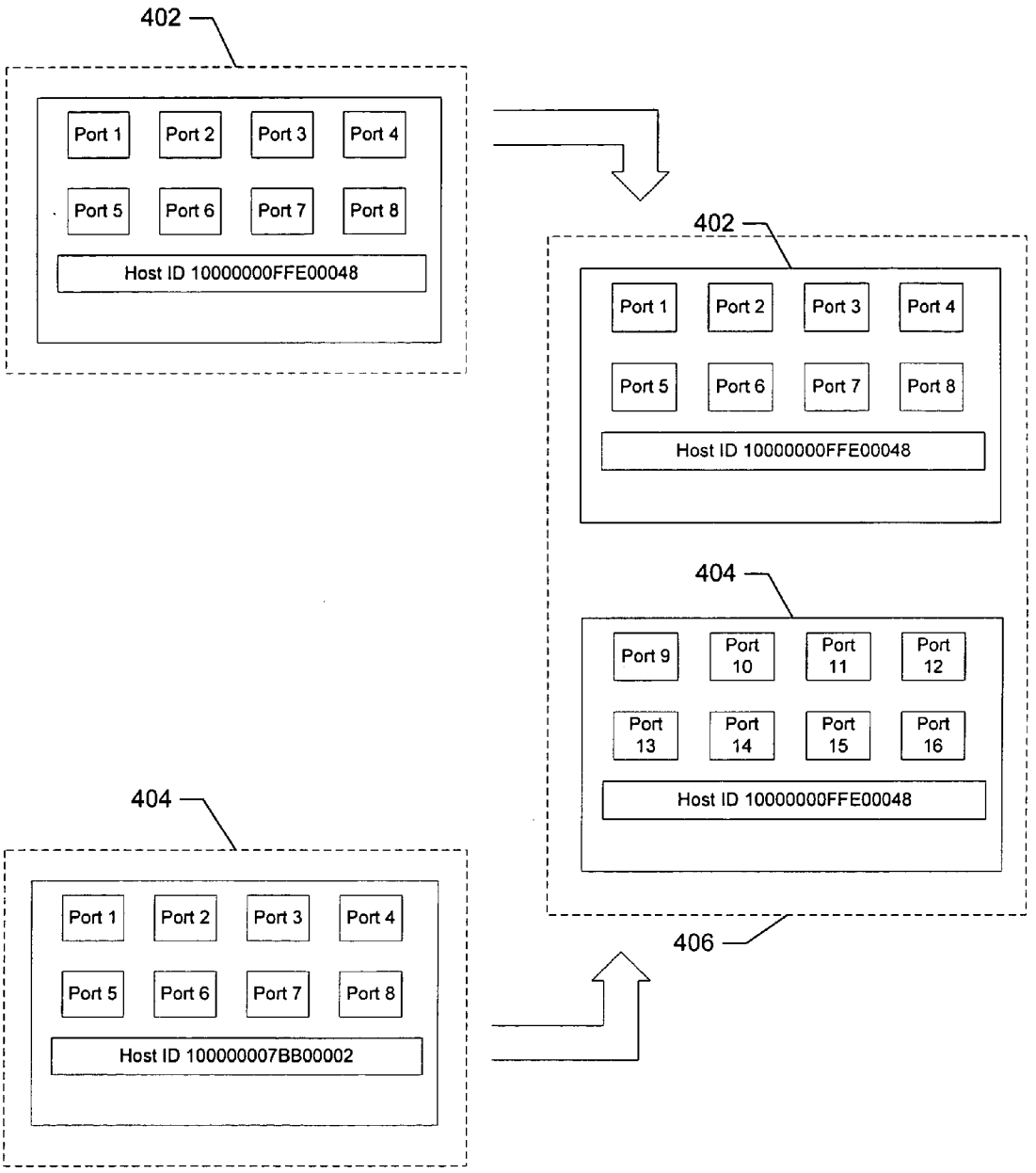

402
Port 1
Port 2
Port 3
Port 4
Port 5
Port 6
Port 7
Port 8
Host ID 10000000FFE00048
402
Port 1
Port 2
Port 3
Port 4
Port 5
Port 6
Port 7
Port 8
Host ID 10000000FFE00048
404
Port 9
Port 10
Port 11
Port 12
Port 13
Port 14
Port 15
Port 16
Host ID 10000000FFE00048
406
404
Port 1
Port 2
Port 3
Port 4
Port 5
Port 6
Port 7
Port 8
Host ID 100000007BB00002

FIG. 4

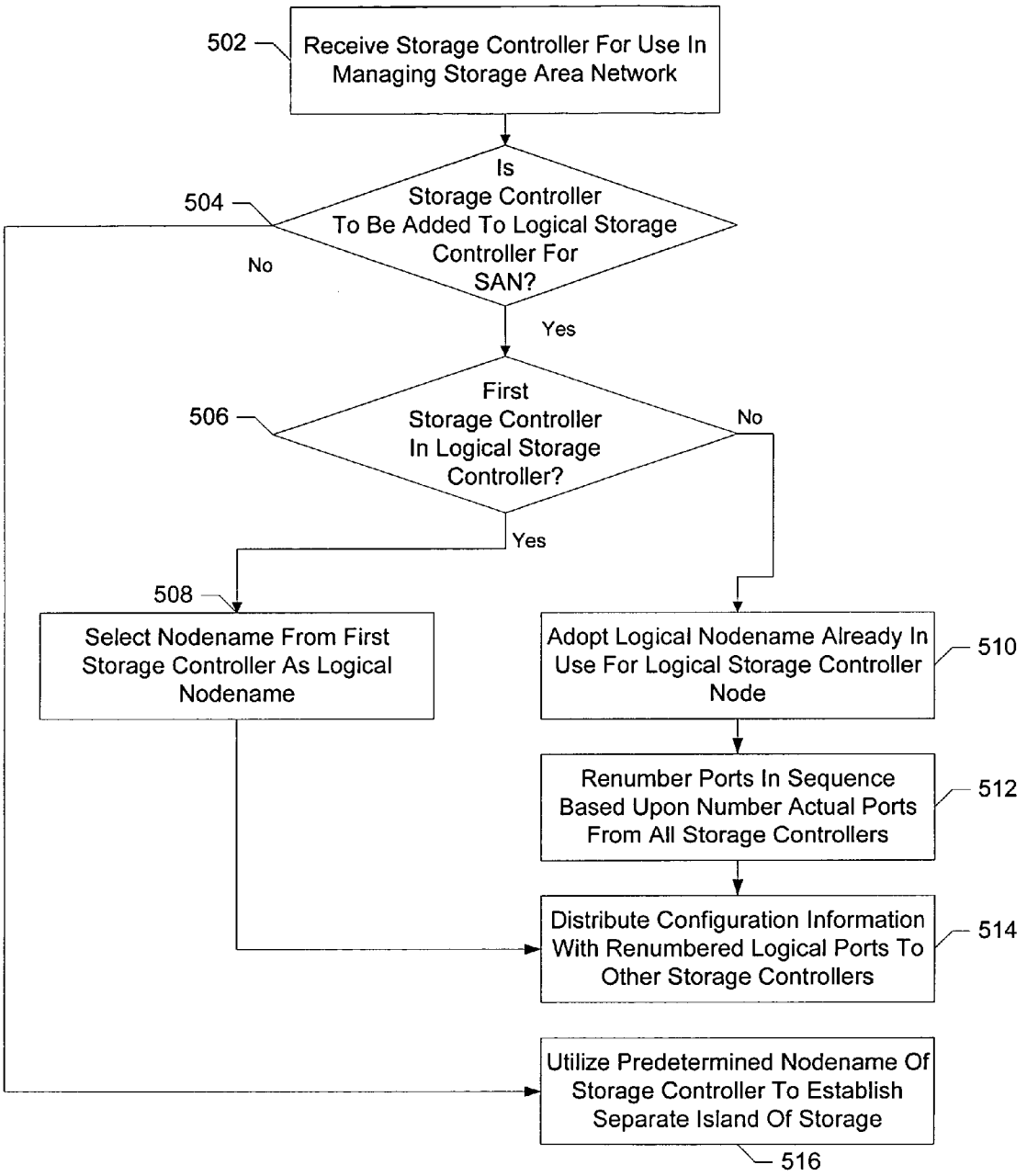

502
Receive Storage Controller For Use In Managing Storage Area Network
504
Is Storage Controller To Be Added To Logical Storage Controller For SAN?
No
Yes
506
First Storage Controller In Logical Storage Controller?
No
Yes
508
Select Nodename From First Storage Controller As Logical Nodename
Adopt Logical Nodename Already In Use For Logical Storage Controller Node
510
Renumber Ports In Sequence Based Upon Number Actual Ports From All Storage Controllers
512
Distribute Configuration Information With Renumbered Logical Ports To Other Storage Controllers
514
Utilize Predetermined Nodename Of Storage Controller To Establish Separate Island Of Storage
516

FIG. 5

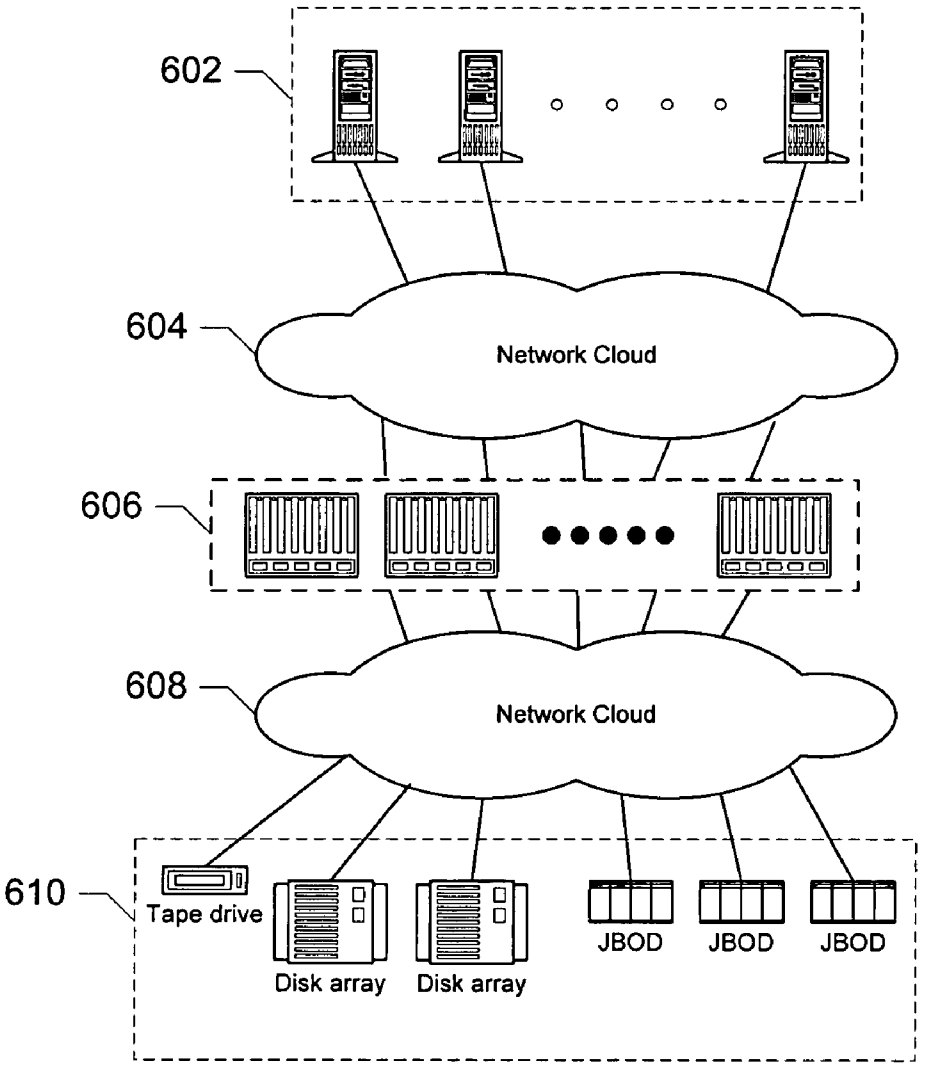

602
604
Network Cloud
606
608
Network Cloud
610
Tape drive
Disk array
Disk array
JBOD
JBOD
JBOD

FIG. 6

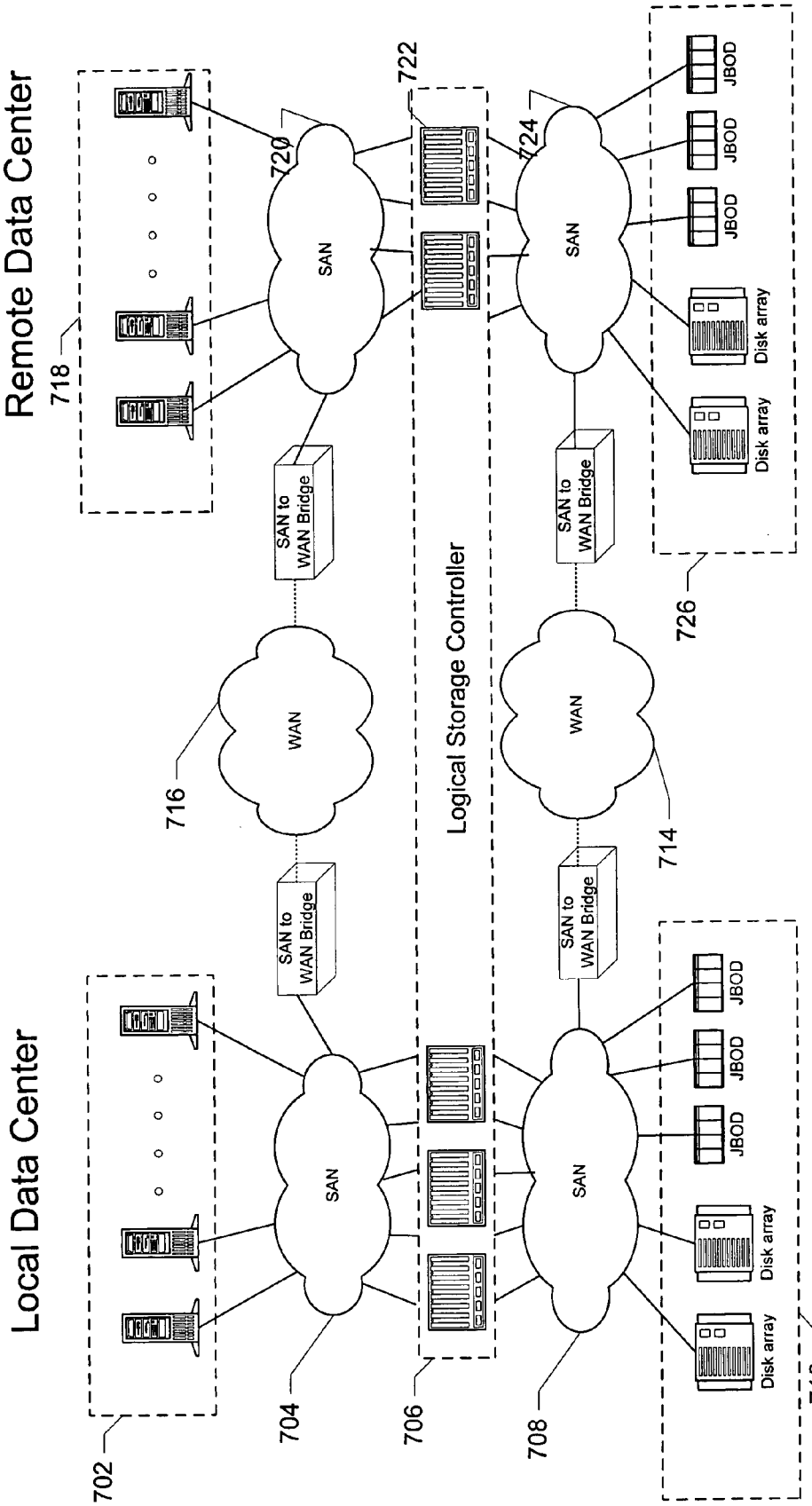

Remote Data Center
718
720
722
724
726
SAN
SAN to WAN Bridge
WAN
716
714
Logical Storage Controller
JBOD
Disk array
Local Data Center
702
704
706
708
710

FIG. 7

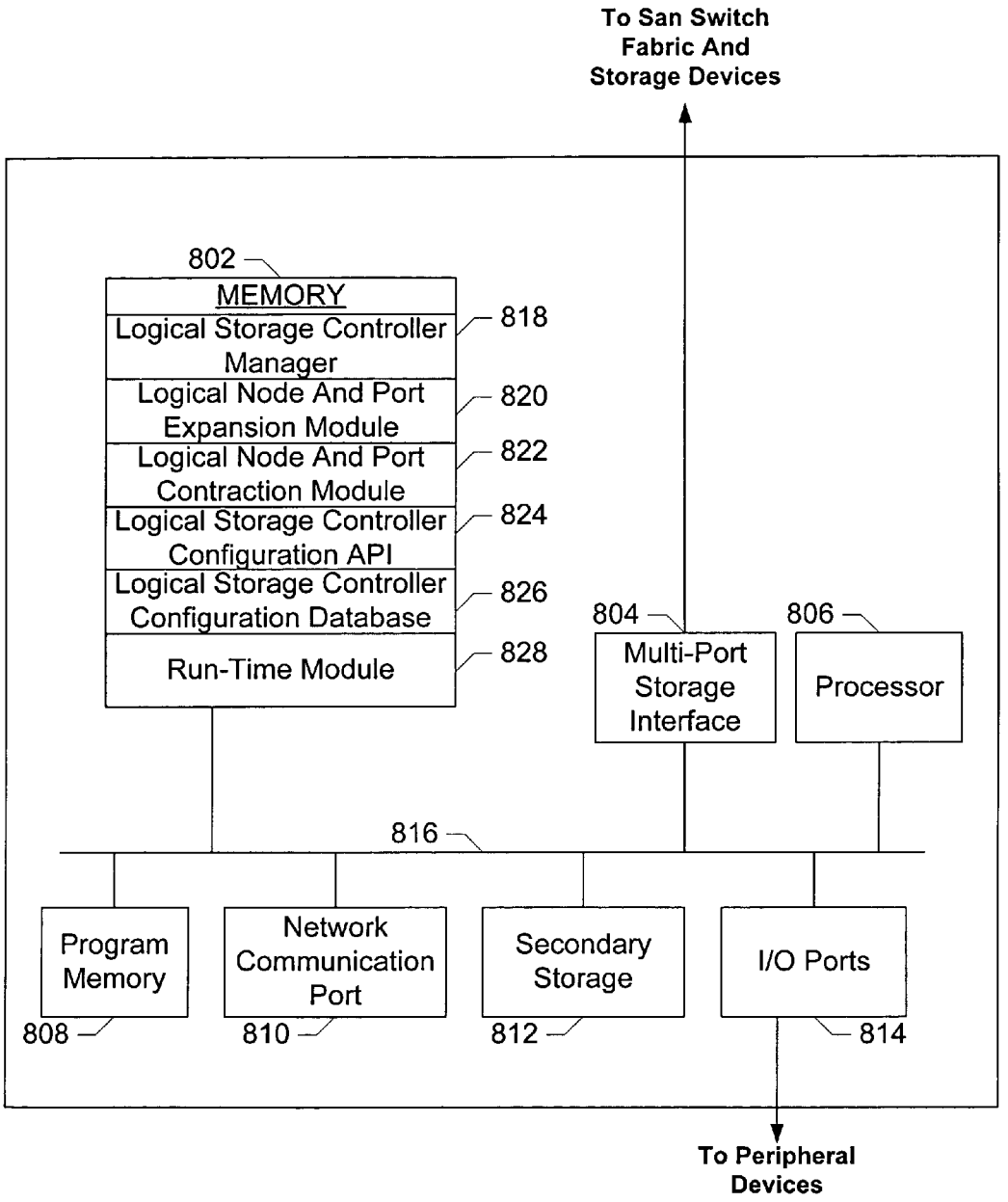

To San Switch Fabric And Storage Devices
802
MEMORY
Logical Storage Controller Manager
818
Logical Node And Port Expansion Module
820
Logical Node And Port Contraction Module
822
Logical Storage Controller Configuration API
824
Logical Storage Controller Configuration Database
826
Run-Time Module
828
804
Multi-Port Storage Interface
806
Processor
816
Program Memory
808
Network Communication Port
810
Secondary Storage
812
I/O Ports
814
To Peripheral Devices

FIG. 8

METHOD AND SYSTEM FOR DYNAMIC EXPANSION AND CONTRACTION OF NODES IN A STORAGE AREA NETWORK

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Provisional Application No. 60/422,109, filed Oct. 28, 2002 and titled "Apparatus and Method for Enhancing Storage Processing in a Network-Based Storage Virtualization System", which is incorporated herein by reference. This application also relates to the subject matter disclosed in the co-pending U.S. application Ser. No. AAAAAAAA (attorney docket 00121-000600000, by Richard Meyer, et al., titled "Method and System for Dynamic Expansion and Contraction of Nodes in a Storage Area Network", co-pending U.S. application Ser. No. BBBBBBBB (attorney docket 00121-0007000000, by Gautam Ghose, et al., titled "Failure Analysis Method and System for Storage Area Networks", co-pending U.S. application Ser. No. CCCCCCCC (attorney docket 00121-0008000000, by Tuan Nguyen, et al., titled "Method and System for Managing Time-Out Events in a Storage Area Network", co-pending U.S. application Ser. No. DDDDDDDD (attorney docket 00121-0009000000, by Rush Manbert, et al., titled "Method and System for Strategy Driven Provisioning of Storage in a Storage Area Network", filed concurrently herewith.

BACKGROUND OF THE INVENTION

[0002] Storage area networks, also known as SANs, facilitate sharing of storage devices with one or more different host server computer systems and applications. Fibre channel switches (FCSs) can connect host servers with storage devices creating a high speed switching fabric. Requests to access data pass over this switching fabric and onto the correct storage devices through logic built into the FCS devices. Host servers connected to the switching fabric can quickly and efficiently share blocks of data stored on the various storage devices connected to the switching fabric.

[0003] Storage devices can share their storage resources over the switching fabric using several different techniques. For example, storage resources can be shared using storage controllers that perform storage virtualization. This technique can make one or more physical storage devices, such as disks, which comprise a number of logical units (sometimes referred to as "physical LUNs") appear as a single virtual logical unit or multiple virtual logical units, also known as VLUNs. By hiding the details of the numerous physical storage devices, a storage virtualization system having one or more such controllers advantageously simplifies storage management between a host and the storage devices. In particular, the technique enables centralized management and maintenance of the storage devices without involvement from the host server.

[0004] In many instances it is advantageous to place the storage virtualization controller(s) in the middle of the fabric, with the host servers and controllers arranged at the outer edges of the fabric. Such an arrangement is generally referred to as a symmetric, in-band, or in-the-data-path configuration. As storage area networks grow additional ports on the storage area controller and switching fabric are needed to connect additional hosts and storage devices into the network. More ports on the storage controller are also required to handle increasing per-port performance requirements. For example, current SANs provide 6,000 ports with a 3:1 fan in/out ratio measure of performance. Assuming the same 3:1 performance, increasing the SAN size by ten-fold to 60,000 ports would require the storage controller device in the SAN to provide over 20,000 ports in the chassis. Even if per-port performance could be increased to a 6:1 fan in/out ratio, the storage controller device would still be required to provide over 10,000 ports.

[0005] Indeed, increasing the size of the storage controller to hold thousands of ports also has drawbacks. Growing customers may not want the added expense of purchasing storage controllers with empty slots to accommodate projected growth and expansion. By the time the company requires the additional ports, advances in technology and manufacturing may render the technology required for the added slots functionally or economically obsolete. Adding ports to a large but older storage controller chassis may be more expensive than buying newer, denser and more reliable storage controller technology. Further, once the larger storage controller is filled then customers must once again face the dilemma of purchasing yet another storage controller chassis and ports. These large storage controllers are not only expensive but cumbersome to manufacture and install.

[0006] Even if it were feasible, larger storage controllers are more vulnerable to hardware or software failures. In a SAN, a single hardware or software failure that renders a physical device inoperable will impact the SAN performance. The significance of the failure may depend on the role of the physical device. For example, in the traditional replication of a storage controller instance failure of an original storage controller may lead to a performance degradation of as much as 50%; in most cases, this is an unacceptable impact on the SAN.

[0007] Current technique for adding another conventional storage controller may satisfy the increased demand for ports but at the expense of significantly increasing the complexity of managing the SAN. Each additional storage controller chassis and set of ports defines a separate island within the overall storage network. These islands have a separate storage pools and therefore cannot share the storage as readily as storage located on a single SAN.

[0008] Performance in the SAN would also most likely suffer when required port ratios in the storage controller described previously are exceeded. Typically performance degradation would create port congestion, over-subscription of resources and reduced input-output performance. The ability to address the increased port demand in the SAN is currently limited by the number of slots and line cards in a given storage controller chassis. Unfortunately, adding additional line cards and ports is not a simple task using conventional storage controller solutions as demonstrated above.

[0009] For these and other reasons, it is therefore desirable to improve the scalability, configurability and sizing of storage controllers as used in rapidly growing SANs.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The features of the present invention and the manner of attaining them, and the invention itself, will be best

understood by reference to the following detailed description of implementations of the invention, taken in conjunction with the accompanying drawings, wherein:

[0011] **FIG. 1** is an exemplary system block diagram of the logical relationship between host servers, storage devices, and a storage area network (SAN) implemented using a switching fabric along with an implementation of the present invention;

[0012] **FIG. 2** is an exemplary system block diagram illustrative of the relationship provided by a storage virtualization controller between virtual logical units and logical units on physical storage devices, in accordance with an implementation of the present invention;

[0013] **FIG. 3A** provides a schematic block diagram illustrating a portion of the physical connections on a storage controller;

[0014] **FIG. 3B** illustrates one scheme used for addressing a storage controller having a nodename and several port names

[0015] **FIG. 4** is a block diagram and overview of combining two storage controllers into one logical storage controller in accordance with implementations of the present invention;

[0016] **FIG. 5** is a flowchart diagramming of the operations used to add a storage controller node in a storage area network in accordance with one implementation of the present invention;

[0017] **FIG. 6** is a block diagram illustrating a SAN using a logical storage controller in accordance with one implementation of the present invention;

[0018] **FIG. 7** is a block diagram system configuration of a local and remote data center operatively coupled together using a logical storage controller designed in accordance with one implementation of the present invention; and

[0019] **FIG. 8** provides one implementation of the present invention as it would be implemented in a computer device or system.

SUMMARY OF THE INVENTION

[0020] In one implementation, the present invention provides a method for adding a storage controller node in a storage area network. Adding the storage controller includes receiving a storage controller node to add to a logical storage controller in the storage area network having a logical nodename and a sequence of logical ports, adopting the logical nodename from the logical storage controller in place of the predetermined nodename associated with the storage controller and renumbering a set of ports associated with the storage controller to extend the sequence of logical ports associated with the logical storage controller.

[0021] In another implementation, the present invention provides a method of removing a storage controller node in a storage area network. Removing a storage controller includes removing a storage controller node from a logical storage controller in the storage area network having a logical nodename and a sequence of logical ports, deleting the set of ports associated with the removed storage controller from the sequence of logical ports associated with the logical storage controller and renumbering the sequence of logical ports to accommodate the deleted set of ports.

DETAILED DESCRIPTION

[0022] Aspects of the present invention provide a dynamic method and system for expanding and contracting nodes in a storage area networks. Managing nodes in accordance with the present invention is advantageous for at least one or more of the following reasons described herein. A single storage pool can be managed with multiple storage controllers configured and managed in accordance with the present invention as a single logical storage controller. Ports can be added without increasing the number of storage islands being managed as ports are added in contiguous sequence with preexisting ports in the logical storage controller. Multiple new storage controllers entered into a logical storage controller contribute additional ports to the SAN while appearing as a single and larger storage controller.

[0023] Implementations of the present invention facilitate combining multiple storage controllers into a single larger logical storage controller. It can also be used to combine existing and older storage controllers with newer storage controllers. The logical storage controller construct implemented in accordance with the present invention accommodates different storage controllers through a universal application programming interface (API). This API is used to exchange database information between the various controllers pertinent to operation of the overall logical storage controller. The API even makes it possible for storage controllers using different databases to share important SAN configuration information and coexist. This API interface also enables customers to upgrade from an older model storage controller to a newer storage controller without having to remove the existing older storage controller model on the SAN.

[0024] Further, another advantage of the present invention provides an increased reliability characteristic given hardware or software failures. In a logical storage controller containing n physical storage controllers the impact of failure from a storage controller can be reduced to approximately 1/n. The larger logical storage controllers have significant improved performance and reliability due to the increased redundancies in ports and communication paths not previously available using conventional solutions.

[0025] Management of remote SAN is also simplified by implementations of the present invention. A remote SAN and storage controller can be incorporated into the logical storage controller of the present invention and managed as a single SAN rather than a local SAN and a remote SAN as traditionally found with other solutions. Also, the storage pool on both the local and remote SAN can be combined into a single shared image or island of storage rather than several.

[0026] Referring to the exemplary configuration in **FIG. 1, a** storage area network (SAN) **100** may include one or more SAN switch fabrics, such as fabrics **104,105**. Fabric **104** is connected to hosts **102**, while fabric **105** is connected to storage devices **106**. At least one storage virtualization controller **126** is inserted in the midst of SAN **100**, and connected to both fabrics **104,105** to form a symmetric, in-band storage virtualization configuration. In an in-band configuration, communications between server devices **102** and storage devices **106** pass through controller **126** for

performing data transfer in accordance with the present invention. In accordance with implementations of the present invention, these one or more storage virtualization controllers **126** or storage controllers **126** appear as a single logical storage controller during operation.

[**0027**] Host servers **102** are generally communicatively coupled (through fabric **104**) via links **150** to individual upstream processing elements (UPE) of storage controller **126**. In an alternate configuration, one or more host servers may be directly coupled to storage controller **126**, instead of through fabric **104**. Storage controller **126** includes at least one UPE for each server **102** (such as host servers **108**,**110**, **112**,**114**) connected to storage controller **126**. As will be discussed subsequently in greater detail, storage controller **126** appears as a virtual logical unit (VLUN) to each host server.

[**0028**] Storage devices **106** are communicatively coupled (through fabric **105**) via links **152** to individual downstream processing elements (DPEs) of storage controller **126**. In an alternate configuration, one or more storage devices may be directly coupled to storage controller **126**, instead of through fabric **105**. Storage controller **126** includes at least one DPE for each storage device **106** (such as storage devices **130**, **132**,**134**,**136**,**138**) connected to the controller **126**. Storage controller **126** appears as an initiator to each storage device **106**. Multiple storage controllers **126** may be interconnected by external communications link **160**. Within each storage controller **126**, is node expansion and node contraction routines for adding and deleting storage controllers with supporting hardware and software needed to implement the present invention. As described later herein, the components used for expanding or contracting the nodes in a logical storage controller facilitate scalability of ports in the storage area network in an efficient cost-effective manner without significant increased management overhead requirements.

[**0029**] Considering now the virtualization of storage provided by an implementation of the present invention, and with reference to the exemplary SAN **200** of **FIG. 2, a** storage virtualization system includes an exemplary storage virtualization controller arrangement **201**. Storage controller arrangement **201** includes, for illustrative purposes, two storage virtualization controllers **202**,**203** interconnected via external communication link **260**. Storage controller**1202** has been configured to provide four virtual logical units **214**,**216**,**218**,**220** associated with hosts **204**-**210**, while controller**2203** has been configured to provide one virtual logical unit **214** associated with hosts **204**,**211**. In the general case, a virtual logical unit (VLUN) includes N "slices" of data from M physical storage devices, where a data "slice" is a range of data blocks. In operation, a host requests to read or write a block of data from or to a VLUN. Through controllers **202** of this exemplary configuration, host**1204** is associated with VLUN**1214**; host**2205**, host**3206**, and host**4207** are associated with VLUN**2216**; host**5208** is associated with VLUN**3218**, host**6209** and host**7210** are associated with VLUN**4220**. Through controller**2203**, host**1204** and host**8211** are also associated with VLUN**1214**. It can be seen that host**1204** can access VLUN**1214** through two separate paths, one through controller**1202** and one path through controller**2203**.

[**0030**] A host **204**-**211** accesses it's associated VLUN by sending commands to the controller arrangement **201** to read and write virtual data blocks in the VLUN. Controller arrangement **201** maps the virtual data blocks to physical data blocks on individual ones of the storage devices **232**, **234**,**236**, according to a preconfigured mapping arrangement. Controller arrangement **201** then communicates the commands and transfers the data blocks to and from the appropriate ones of the storage devices **232**,**234**,**236**. Each storage device **232**,**234**,**236** can include one or more physical LUNs; for example, storage device **1232** has two physical LUNs, LUN **1**A **222** and LUN **1**B **223**.

[**0031**] To illustrate further the mapping of virtual data blocks to physical data blocks, all the virtual data blocks of VLUN**1214** are mapped to a portion **224***a* of the physical data blocks LUN**2224** of storage device **234**. Since VLUN**2216** requires more physical data blocks than any individual storage device **232**,**234**,**236** has available, one portion **216***a* of VLUN**2216** is mapped to the physical data blocks of LUN**1**A **222** of storage device **232**, and the remaining portion **216***b* of VLUN**2216** is mapped to a portion **226***a* of the physical data blocks of LUN**3226** of storage device **236**. One portion **218***a* of VLUN**3218** is mapped to a portion **224***b* of LUN**2224** of storage device **234**, and the other portion **218***b* of VLUN**3218** is mapped to a portion **226***b* of LUN**3226** of storage device **236**. It can be seen with regard to VLUN**3** that such a mapping arrangement allows data block fragments of various storage devices to be grouped together into a VLUN, thus advantageously maximizing utilization of the physical data blocks of the storage devices. All the data blocks of VLUN**4220** are mapped to LUN**1**B **223** of storage device **232**.

[**0032**] While the above-described exemplary mapping illustrates the concatenation of data block segments on multiple storage devices into a single VLUN, it should be noted that other mapping schemes, including but not limited to striping and replication, can also be utilized by the controller arrangement **201** to form a VLUN. Additionally, the storage devices **232**,**234**,**236** may be heterogeneous; that is, they may be from different manufacturers or of different models, and may have different storage sizes, capabilities, architectures, and the like. Similarly, the hosts **204**-**210** may also be heterogeneous; they may be from different manufacturers or of different models, and may have different processors, operating systems, networking software, applications software, capabilities, architectures, and the like.

[**0033**] It can be seen from the above-described exemplary mapping arrangement that different VLUNs may contend for access to the same storage device. For example, VLUN**2216** and VLUN**4220** may contend for access to storage device **1232**; VLUN**1214** and VLUN**3218** may contend for access to storage device **2234**; and VLUN**2216** and VLUN**3218** may contend for access to storage device **3236**. The storage virtualization controller arrangement **201** according to an implementation of the present invention performs the mappings and resolves access contention, while allowing data transfers between the host and the storage device to occur at wire-speed.

[**0034**] Before considering the various elements of the storage virtualization system in detail, it is useful to discuss, with reference to **FIGS. 1 and 2**, the format and protocol of the storage requests that are sent over SAN **200** from a host to a storage device through the controller arrangement **201**. Many storage devices frequently utilize the Small Computer

System Interface (SCSI) protocol to read and write the bytes, blocks, frames, and other organizational data structures used for storing and retrieving information. Hosts access a VLUN using these storage devices via some implementation of SCSI commands; for example, layer **4** of Fibre Channel protocol. However, it should be noted that the present invention is not limited to storage devices or network commands that use SCSI protocol.

[**0035**] Storage requests may include command frames, data frames, and status frames. The controller arrangement **201** processes command frames only from hosts, although it may send command frames to storage devices as part of processing the command from the host. A storage device generally does not send command frames to the controller arrangement **201**, but instead sends data and status frames. A data frame can come from either host (in case of a write operation) or the storage device (in case of a read operation).

[**0036**] In many cases one or more command frames is followed by a large number of data frames. Command frames for read and write operations include an identifier that indicates the VLUN that data will be read from or written to. A command frame containing a request, for example, to read or write a 50 kB block of data from or to a particular VLUN may then be followed by 25 continuously-received data frames each containing 2 kB of the data. Since data frames start coming into the controller **203** only after the controller has processed the command frame and sent a go-ahead indicator to the host or storage device that is the originator of the data frames, there is no danger of data loss or exponential delay growth if the processing of a command frame is not done at wire-speed; the host or the storage device will not send more frames until the go-ahead is received. However, data frames flow into the controller **203** continuously once the controller gives the go-ahead. If a data frame is not processed completely before the next one comes in, the queuing delays will grow continuously, consuming buffers and other resources. In the worst case, the system could run out of resources if heavy traffic persists for some time.

[**0037**] **FIG. 3A** provides a schematic block diagram illustrating a portion of the physical connections on a storage controller. In this example illustration, storage controller **302** includes 8 ports as indicated and is identified with a predetermined nodename determined during manufacturing. This predetermined nodename is also referred to as a world wide nodename (node WWN) and is designed to be distinguishable on a network according to the manufacture and the production sequence of the storage controller by the manufacture. Given these designations, a port on each storage controller brand new out of the box is uniquely identified as a combination of the port and predetermined nodename associated with the storage controller.

[**0038**] **FIG. 3B** illustrates one scheme used for addressing a storage controller having a nodename and several port names. In **FIG. 3**B, the nodename is derived from a media access control address (MAC) **306** defined using 48 bits number **0-47**. Company specific **314** in bits **0-23** define a vendor specified identifier created when the company manufacturers each particular product. IEEE Company Code in bits **24-47** contains an identifier assigned to each company producing or manufacturing the product. MAC **306** can also be used in association with an Ethernet port in addition to using it to construct a nodename or WWN. As will be described later herein, implementations of the present invention may alter both the predetermined IEEE Company Code **312** as well as the predetermined Company Specific code **314** (together making up the "nodename") so as to change the definition of the values described previously.

[**0039**] Depending on the values contained within, a 64-bit string can describe a nodename **308** or a portname **310**. In the first instance, a nodename **308** is indicated by a "0×01" value in bits **60-63** of field Network Address Authority (NAA) **316** ID and a "0" value in bits **48-59** of Port ID **317**. Along with IEEE Address **318** or MAC as described above, nodename **308** identifies a WWN for the storage controller. Alternatively, portname **310** is indicated by a "0×02" value in bits **60-63** of field NAA **320** and a port number in bits **48-59** of Port ID **321** identifying the particular port. The port number in bits **48-59** corresponds to the port being identified on the node. IEEE Address **322** in bits **0-47** once again includes a MAC address to identify the storage controller.

[**0040**] Referring to **FIG. 4** is a block diagram and overview of combining two storage controllers into one logical storage controller in accordance with implementations of the present invention. In this example, a first storage controller **402** has 8 ports identified as port **1** through port **8** and a predetermined node name of "10000000FFE00048" assigned to the storage controller during manufacture. Similarly, a second storage controller **404** also has 8 ports ranging from port **1** to port **8** but instead has a predetermined nodename of "100000007BB00002", also assigned during manufacture.

[**0041**] In accordance with the present invention, first storage controller **402** is entered into a logical storage controller arrangement **406**. Subsequently, storage controller **404** is added to the arrangement to be used in conjunction with first storage controller **402**. In this particular implementation, a logical nodename of "10000000FFE00048" is derived from the predetermined nodename assigned to first storage controller **402** as this storage controller was also the first storage controller added to logical storage controller arrangement **406**. As illustrated, second storage controller **304** adopts the logical nodename of "10000000FFE00048" assigned to the logical storage controller arrangement **406** and renumbers the ports in contiguous sequence extending the logical ports of logical storage controller **406** from 8 to a total of 16.

[**0042**] **FIG. 5** is a flowchart diagramming of the operations used to add a storage controller node in a storage area network in accordance with one implementation of the present invention. Initially, a storage controller is received for use in assisting in the managing of the storage area network and expanding the number of available ports **(502)**. The storage controller received has a predetermined nodename and set of ports defined during manufacture.

[**0043**] The administrator or other party configuring the system determines if the storage controller is to be added to a logical storage controller for the SAN or work as a separate storage controller **(504)**. Like conventional systems, implementations of the present invention provide the option of adding the storage controller without further modification using the predetermined nodename of the storage controller and set of ports to establish a separate island of storage **(516)**.

[0044] Alternatively, a determination is made if the storage controller is the first storage controller in the logical storage controller (**506**). In one implementation, the first storage controller added to the logical storage controller arrangement contributes a nodename for use as the logical nodename of the logical storage controller. The logical nodename is a unique world wide node name (WWN).

[0045] When at least one storage controller is associated with the logical storage controller arrangement (**506**), the storage controller adopts the logical nodename from the logical storage controller in place of the predetermined nodename for the storage controller determined during manufacture (**510**). In addition, implementations of the present invention renumber the set of ports associated with the storage controller to extend the sequence of logical ports associated with the logical storage controller (**512**). For example, in one configuration the sequence of logical ports is a contiguous numeric sequence of ports generated as sets of ports from each storage controller as they are added to the logical storage controller.

[0046] Implementations of the present invention also designate a role for each storage controller in the logical storage controller including: a primary storage controller, a secondary storage controller and a alternate storage controller. The primary storage controller is generally responsible for a variety of roles involved with updating data across the storage controllers in the logical storage controller. In the event the primary storage controller experiences a failure, the secondary storage controller performs tasks assigned to the primary storage controller; alternate storage controllers can also be nominated to step up to a secondary storage controller role as deemed necessary. Once the storage controller is modified, the present invention then generates configuration information reflecting the addition of the storage controller to the logical storage controller and the set of ports added to the corresponding sequence of logical ports.

[0047] Implementations of the present invention then distributes configuration information containing the above changes to one or more storage controllers associated with the logical storage controller (**514**). In one implementation, the configuration information is stored in a database along with other types of metadata useful in operating the logical storage controller of the present invention. For example, the metadata describing the geometry of one or more volumes of data managed by the logical storage controller may include number of physical devices, physical device identifiers, ranges of blocks on the physical devices and total number of blocks. To maintain compatibility, an application programming interface (API) of the present invention passes this metadata and configuration information to one or more databases associated with the respective storage controllers. The API of the present invention communicates with many different types of databases and thereby facilitates combining many different types of storage controllers. Older storage controllers communicate with newer storage controllers and different model storage controllers over an external communication link using this API of the present invention to pass this information.

[0048] Aspects of the present invention can also be used to reduce the size of nodes in a logical storage controller. Removing a storage controller node from a logical storage controller in the storage area network affects the number of ports and the sequence of ports. The set of ports associated with the removed storage controller are deleted from the sequence of logical ports associated with the logical storage controller and renumbered to reverse out the operations performed when they were added. The storage controller removed from the logical storage controller cannot be used in another SAN unless it is also properly reprogrammed with a different nodename and set of ports.

[0049] Other storage controllers must also be notified of the changes when the storage controller is removed. Accordingly, implementations of the present invention generates configuration information except this time reflecting the deleted storage controller removed from the logical storage controller and the set of ports removed from the corresponding sequence of logical ports. This configuration information and metadata is also once again distributed to one or more storage controllers still associated with the logical storage controller.

[0050] **FIG. 6** is a block diagram illustrating a SAN using a logical storage controller in accordance with one implementation of the present invention. As illustrated, a SAN includes hosts **602**, a switching fabric **604**, a logical storage controller **606** configured in accordance with the present invention, a switching fabric **608** and a storage servers **610**. In operation, hosts **602** appear to be accessing a single storage controller namely logical storage controller **606** and therefore have shared access to a single pool of storage servers **610** rather then multiple islands of storage as provided in conventional solutions.

[0051] Further, logical storage controller **606** is more reliable as each storage controller within the logical storage controller operates as a backup when ports or whole storage controllers fail. Failed ports or storage controller details are passed between the storage controllers in logical storage controller **606** to create alternate paths between hosts **602** and storage servers **610** thus keeping the SAN operable. If properly configured, a large number of storage controllers n in logical storage controller **606** limits the impact of a fault or failure of hardware or software to 1/n rather than a higher impact suffered in conventional systems.

[0052] Referring to **FIG. 7** is a block diagram system configuration of a local and remote data center operatively coupled together using a logical storage controller designed in accordance with one implementation of the present invention. Local data center includes hosts **702**, a switching fabric **704**, a logical storage controller **706** portion designed in accordance with the present invention, a fabric **708** and storage servers **710**. Similarly, remote data center includes hosts **718**, a switching fabric **720**, a logical storage controller **722** portion designed in accordance with the present invention, a fabric **724** and storage servers **726**. Wide area network and equipment **716** and wide area network and equipment **714** couple the local data center and remote data center together over a distance. In this example, storage servers **710** and storage servers **726** are physically separate but appear in the same storage pool from both hosts **702** and hosts **718**. In large enterprises spanning countries or even continents, this type of arrangement would greatly simplify SAN data storage management as well as improve sharing of underlying data stored in these systems.

[0053] **FIG. 8** provides one implementation of the present invention as it would be implemented in a computer device

or system. In this example, system **800** includes a memory **802**, typically random access memory (RAM), a multiport storage interface **804**, a processor **806**, a program memory **808** (for example, a programmable read-only memory (ROM) such as a flash ROM), a network communication port **810** as an alternate communication path, a secondary storage **812**, and I/O ports **814** operatively coupled together over interconnect **816**. System **800** can be preprogrammed, in ROM, for example using a microcode or it can be programmed (and reprogrammed) by loading a program from another source (for example, from a floppy disk, a CD-ROM, or another computer) and preferably operates using real-time operating system constraints.

[**0054**] Memory **802** includes various components useful in implementing aspects of the present invention. These components include a logical storage controller manager **818**, logical node and port expansion module **820**, a logical node and port contraction module **822**, a logical node configuration API **824** and a logical node configuration database **826** managed using a run-time module **828**.

[**0055**] Logical storage controller manager **818** manages the operation of the logical storage controller and communicates information about the status of the logical storage controller to one or more underling storage controllers. Some of these operations may be delegated and distributed to primary, secondary or alternate storage controllers associated with the logical storage controller. Higher level operations handled by logical storage controller manager includes mapping failed ports to functioning ports on other devices as well as redirecting SAN traffic away from failed storage controllers in the logical storage controller.

[**0056**] Logical node and port expansion module **820** performs operations on storage controllers added to the logical storage controller and updates various configuration information, metadata and databases as needed to keep the logical storage controller and system operational. Further, logical node and port contraction module **822** handles the removal of certain storage controllers from the logical storage controller network and the renumbering of ports as needed and previously described.

[**0057**] Logical storage controller configuration API (application programming interface) **824** provides the interface used to communicate between the different storage controllers. This API facilitates communication with a wide variety of databases and thus keeps legacy systems compatible with newer systems as well as with systems manufactured by different third-parties. Logical storage controller configuration database **826** is one example database that holds configuration information, metadata and other information about the current status of the SAN implemented in accordance with a logical storage controller of the present invention. Resources used to run these aforementioned modules are managed efficiently using a run-time module **828** for higher performance and a small memory footprint.

[**0058**] In general, implementations of the invention can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Apparatus of the invention can be implemented in a computer program product tangibly embodied in a machine readable storage device for execution by a programmable processor; and method steps of the invention can be performed by a programmable processor executing a program of instructions to perform functions of the invention by operating on input data and generating output. The invention can be implemented advantageously in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program can be implemented in a high level procedural or object oriented programming language, or in assembly or machine language if desired; and in any case, the language can be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, the processor receives instructions and data from a read only memory and/or a random access memory. Also, a computer will include one or more secondary storage or mass storage devices for storing data files; such devices include magnetic disks, such as internal hard disks and removable disks; magneto optical disks; and optical disks. Storage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, such as EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto optical disks; and CD ROM disks. Any of the foregoing can be supplemented by, or incorporated in, ASICs (application specific integrated circuits).

[**0059**] While specific implementations have been described herein for purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. Accordingly, the invention is not limited to the above-described implementations, but instead is defined by the appended claims in light of their full scope of equivalents. From the foregoing it will be appreciated that the storage virtualization controller arrangement, system, and methods provided by the present invention represent a significant advance in the art. Although several specific implementations of the invention have been described and illustrated, the invention is not limited to the specific methods, forms, or arrangements of parts so described and illustrated. For example, the invention is not limited to storage systems that use SCSI storage devices, nor to networks utilizing fibre channel protocol. This description of the invention should be understood to include all novel and non-obvious combinations of elements described herein, and claims may be presented in this or a later application to any novel and non-obvious combination of these elements. The foregoing implementations are illustrative, and no single feature or element is essential to all possible combinations that may be claimed in this or a later application. Unless otherwise specified, steps of a method claim need not be performed in the order specified. The invention is not limited to the above-described implementations, but instead is defined by the appended claims in light of their full scope of equivalents. Where the claims recite "a" or "a first" element of the equivalent thereof, such claims should be understood to include incorporation of one or more such elements, neither requiring nor excluding two or more such elements.

What is claimed is:

1. A method for adding a storage controller node in a storage area network, comprising:

receiving a storage controller node to add to a logical storage controller in the storage area network having a logical nodename and a sequence of logical ports;

adopting the logical nodename from the logical storage controller in place of the predetermined nodename associated with the storage controller; and

renumbering a set of ports associated with the storage controller to extend the sequence of logical ports associated with the logical storage controller.

2. The method of claim 1 further comprising:

generating configuration information reflecting the additional storage controller added to the logical storage controller and the set of ports added to the corresponding sequence of logical ports; and

distributing the configuration information to one or more storage controllers associated with the logical storage controller.

3. The method of claim 1 wherein the logical nodename associated with the logical storage controller is derived from a predetermined nodename associated with one storage controller.

4. The method of claim 1 wherein the predetermined nodename from the first storage controller added to the logical storage controller is used for the logical nodename.

5. The method of claim 1 wherein the logical nodename is a unique world wide node name (WWN).

6. The method of claim 1 wherein the sequence of logical ports is a contiguous numeric sequence of ports generated as sets of ports from each storage controller are added to the logical storage controller.

7. The method of claim 1 wherein each storage controller in the logical storage controller communicates with each other over an external communication link.

8. The method of claim 1 wherein each storage controller added to the logical storage controller is designated a role selected from a set of roles including: a primary storage controller, a secondary storage controller and a alternate storage controller.

9. The method of claim 8 wherein the secondary storage controller performs tasks assigned to the primary storage controller when the primary storage controller experiences a failure.

10. The method of claim 2 wherein the configuration information generated includes metadata describing the geometry of one or more volumes of data managed by the logical storage controller.

11. The method of claim 10 wherein the metadata information is selected from a set including: number of physical devices, physical device identifiers, ranges of blocks on the physical devices and total number of blocks.

12. The method of claim 2 wherein distributing the configuration information is performed using an application programming interface compatible with one or more databases.

13. A method of removing a storage controller node in a storage area network, comprising:

removing a storage controller node from a logical storage controller in the storage area network having a logical nodename and a sequence of logical ports;

deleting the set of ports associated with the storage controller removed from the sequence of logical ports associated with the logical storage controller; and

renumbering the sequence of logical ports to accommodate the deleted set of ports.

14. The method of claim 1 further comprising:

generating configuration information reflecting the deleted storage controller removed from the logical storage controller and the set of ports removed from the corresponding sequence of logical ports; and

distributing the configuration information to one or more storage controllers associated with the logical storage controller.

15. An apparatus for adding a storage controller node in a storage area network, comprising:

a processor capable of executing instructions;

a memory containing instructions when executed on the processor receive a storage controller node to add to a logical storage controller having a logical nodename and a sequence of logical ports in the storage area network, adopt the logical nodename from the logical storage controller in place of the predetermined nodename associated with the storage controller and renumber a set of ports associated with the storage controller to extend the sequence of logical ports associated with the logical storage controller.

16. The apparatus of claim 15 further comprising instructions that generate configuration information reflecting the additional storage controller added to the logical storage controller and the set of ports added to the corresponding sequence of logical ports and distribute the configuration information to one or more storage controllers associated with the logical storage controller.

17. The apparatus of claim 15 wherein the instructions derive the logical nodename associated with the logical storage controller from a predetermined nodename associated with one storage controller.

18. The apparatus of claim 15 wherein the instructions use the predetermined nodename from the first storage controller added to logical storage controller for the logical nodename

19. The apparatus of claim 15 wherein the logical nodename is a unique world wide node name (WWN).

21. The apparatus of claim 15 wherein the sequence of logical ports is a contiguous numeric sequence of ports generated as sets of ports from each storage controller are added to the logical storage controller.

22. The apparatus of claim 15 wherein instructions in each storage controller in the logical storage controller communicate with each other over an external communication link.

23. The apparatus of claim 15 wherein instructions designate a role to each storage controller added to the logical storage controller selected from a set of roles including: a primary storage controller, a secondary storage controller and a alternate storage controller.

24. The apparatus of claim 23 wherein the secondary storage controller performs tasks assigned to the primary storage controller when the primary storage controller experiences a failure.

25. The apparatus of claim 16 wherein the instructions that generate the configuration information includes metadata describing the geometry of one or more volumes of data managed by the logical storage controller.

26. The apparatus of claim 16 wherein instructions selected the metadata information from a set including: number of physical devices, physical device identifiers, ranges of blocks on the physical devices and total number of blocks.

27. The apparatus of claim 16 wherein instructions perform distribution of the configuration information using an application programming interface compatible with one or more databases.

28. An apparatus for removing a storage controller node in a storage area network, comprising:

a processor capable of executing instructions;

a memory containing instructions when executed on the processor remove a storage controller node from a logical storage controller in the storage area network having a logical nodename and a sequence of logical ports, delete the set of ports associated with the storage controller removed from the sequence of logical ports associated with the logical storage controller and renumber the sequence of logical ports to accommodate the deleted set of ports.

29. The apparatus of claim 1 further comprising instructions that generate configuration information reflecting the deleted storage controller removed from the logical storage controller and the set of ports removed from the corresponding sequence of logical ports and distribute the configuration information to one or more storage controllers associated with the logical storage controller.

* * * * *